

# IN1010 V21, Obligatorisk oppgave 7

Innleveringsfrist: 10. mai kl 23:59 (NB: 3 uker)

Versjon 1.1

## Innledning

I denne oppgaven skal du bruke Java med AWT+Swing for å lage et grafisk brukergrensesnitt (GUI) for labyrintprogrammet du lagde i obligatorisk oppgave 6. Du kan godt endre løsningen du leverte der du synes det er fornuftig. Før du begynner å programmere bør du legge en plan for hvordan brukergrensesnittet kan bygges opp på en god måte både med tanke på logikk og visuelt design. Det kan være nyttig å tegne noen utkast for hånd.

Hvis du trenger noen tips til hvordan du skal bygge opp programmet ditt, kan du se nederst under «Tips til løsning av Oblig 7».

## Del A: Presentasjon av utvei

Skriv om programmet slik at du bruker klassen *JFileChooser* for å la brukeren finne/velge filen med labyrinten.

Når brukeren har valgt labyrintfilen som skal åpnes, skal labyrinten vises grafisk. For den grafiske representasjonen av labyrinten skal programmet bruke klassen *GridLayout*.

Når brukeren klikker på en hvit rute, skal du bruke programmet fra oblig 6 for å finne alle løsningene fra denne ruten. Du skal etterpå ta én løsning fra beholderen med løsninger og vise denne i labyrinten. Er det flere enn én løsning, kan du vise den du vil, med informasjon om hvor mange løsninger som totalt ble funnet.

**Relevante Trix-oppgaver:** [Alle oppgaver om GUI.](#)

## Oppsummering

Du skal levere all Java-kode som trengs for at hovedprogrammet skal fungere, inkludert nødvendige klasser fra oblig 6. Du kan ha én fil for hver klasse eller alle klassene i én fil.

Unngå bruk av packages (spesielt relevant ved bruk av IDE-er som IntelliJ). *Ikke lever ZIP-filer!* (Det går an å laste opp flere filer samtidig i Devilry.)

**Merk:** Resten av oppgaven er frivillig.

## Del B (valgfri, men sterkt anbefalt): Vise flere utveier

Om det er flere løsninger, er det nyttig å kunne se alle løsningene av labyrinten. Utvid GUI-et med knapper slik at løsningene kan vises én etter én når brukeren trykker på en knapp. I tillegg til å vise antall løsninger kan det også være lurt å vise hvilket løsningsnummer som presenteres akkurat nå.

## Tips til løsning av oblig 7

Dette er en fornuftig rekkefølge for utviklingen av programmet. Kompiler og kjør hver del for å sikre at den fungerer som den skal. Legg inn testutskrifter for å sjekke det.

1. Som alltid: Les **hele oppgaveteksten** nøye før du begynner å programmere.
2. Start med GUI-delen av koden, for den er vanskeligst å endre siden (om du skulle gjøre noe galt).
3. Ta utgangspunkt i et fungerende program som ligner, for eksempel TrippTrappTresko fra forelesningen.
4. Legg først inn koden som oppretter GUI-vinduet; anta en fast størrelse på labyrinten (f.eks. 12 x 12). Bruk en *GridLayout*, og la hver rute være en *JButton* eller en subklasse av *JButton*.
5. Legg så inn håndtering av hendelser. La brukerens klikk i en rute gi en utskrift, for eksempel rutens kolonne- og radnummer.
6. Skriv koden som får tak i labyrintfilens navn; bruk en *JFileChooser*.
7. Legg inn delen fra oblig 6 som leser inn labyrinten fra filen. Endre koden i punkt 4 til å vise en labyrint av riktig størrelse.
8. Hvordan vise de ulike rutene i labyrinten (hvit, sort, utgang og (i punkt 10) løsningsveien)? Forslag: hvit og utgang er blank (dvs " "), sort er "#" og løsningsveien er "X". (Senere kan du legge til farger om du vil.)
9. Koble sammen GUI-koden med labyrintløseren fra oblig 6. La klikk på en hvit rute i labyrinten starte beregningen av alle løsningsveiene fra den ruten. Velg én av de genererte løsningsveiene og skriv den ut.
10. Marker alle rutene på den genererte løsningsveien; bruk metoden *setText* i alle de aktuelle *JButton*-ene.
11. Endre punkt 9 slik at den blanker alle rutene i løsningsveien før en ny løsning genereres.

# Obligatoriske oppgaver og andre innleveringer ved Ifi

## Spesifikke retningslinjer for obligatorisk aktivitet/undervisning under korona-situasjonen for studenter ved MN-fakultetet våren 2020-21.

### Resultat av studentens innsats

Ved alle pålagte innleveringer av oppgaver ved Ifi, enten det dreier seg om obligatoriske oppgaver, hjemmeeksamen eller annet, kreves det at arbeidet er et resultat av studentens egen innsats.

### Krav til innleverte oppgaver

Å utgi andres arbeid for sitt eget er både ulovlig og uetisk og vil medføre sterke reaksjoner fra IFIs og Universitetets side, for eksempel utvisning i ett eller flere semestre; se [Rutiner for behandling av fuskesaker](#).

Vær derfor oppmerksom på:

Hvis du tar med tekst, programkode, illustrasjoner og annet som andre har laget, må du tydelig merke det og angi hvor det kommer fra.

Det er greit å få hint om hvordan en oppgave kan løses, men dette skal eventuelt brukes som grunnlag for egen løsning og ikke kopieres uendret inn.

Du kan bli innkalt til samtale om dine innleveringer. Du må da kunne forklare innholdet i detalj og redegjøre for hvorledes det innleverte arbeidet er blitt til. Dersom samtalen avdekker at oppgaven ikke er et selvstendig arbeid og/eller du ikke kan gjøre rede for det innleverte arbeidet ditt, kan faglærer velge å ikke godkjenne oppgaven, også selv om gruppelærer har godkjent oppgaven i forkant av samtalen.

## Gruppearbeid

I noen emner skal det leveres gruppearbeid. IFI krever da at alle medlemmer av gruppen kan gjøre rede for hovedtrekkene i det innleverte arbeidet. Dessuten må alle ha utført en rimelig del av det hele, og kunne identifisere og gjøre rede for i detalj sin del.

## Samarbeid ved individuelle innleveringer

Disse kravene betyr ikke at IFI fraråder samarbeid – tvert imot. IFI oppfordrer studentene til å utveksle faglige erfaringer. I tilfeller hvor det skal leveres individuelle oppgaver er følgende viktig:

Du kan diskutere en løsning sammen med andre, men dere skal ikke dele noen deler av løsningen (f.eks ved å levere inn lik kode hvor kun variabelnavn er byttet ut).

Dersom du tar med tekst, programkode, illustrasjoner, og annet som andre har laget, må du tydelig merke det og angi hvor det kommer fra - i en selvstendig oppgave er dette noe som sjelden skal forekomme.

Hvis du er i tvil om hva som er lovlig samarbeid, må du kontakte gruppelærer eller faglærer.

## Retningslinjer for obligatoriske oppgaver

En obligatorisk oppgave er en oppgave som må besvares og godkjennes for å kunne gå opp til avsluttende eksamen i et emne. Besvarelsen på en obligatorisk oppgave vurderes til godkjent eller ikke godkjent og kan ikke inngå i karaktervurderingen i emnet.

Antall obligatoriske oppgaver som gis i et emne samt tidspunktet for offentliggjøring og innlevering av oppgavene skal være klart på semestersiden for emnet ved studiestart.

Av oppgaveformuleringen skal det gå tydelig frem hva som forventes av innleveringen. Den faglige bakgrunnen som er nødvendig for å kunne løse oppgaven må være forelest eller gjort tilgjengelig på annen måte i god tid før innleveringsfristen.

[Retningslinjene i sin helhet](#)

Kontakt Studieadministrasjonen på IFI

E-post: [studieinfo@ifi.uio.no](mailto:studieinfo@ifi.uio.no)